マルチメディア通信と分散処理 92－9
（1999．2．26）

仮想ブロードキャストリンクを利用した片方向通信路の透過的経路制御

藤枝俊輔, 渡部陽仁, 楠本博之
慶應義塾大学

あらまし 本稿では、片方向の通信路を含むネットワークにおける経路制御の問題を解決するため、片方向通信路上に仮想ブロードキャストリンクを構築する手法を述べる。また、LAN上でのシミュレーションによりその有効性を示す。片方向通信路をインターネットで利用する場合、既存の経路制御技術が通信媒体の双方向性を前提とすることが問題となる。その例として、片方向通信路を含むネットワーク上で動的な経路制御プロトコルが正しく動作しないことがあげられる。本機構ではトンネリングを用いて片方向通信路の復路となる接続性を提供する。また、片方向通信路に受信専用で接続するノードが生成したデータリンクパケットを送信機能を持つノードを通して伝送し、通常のブロードキャストリンクと同等の機能を実現する。

Transparent Routing in the Network with Unidirectional Links Using Virtual Broadcast Links

Shunsuke Fujieda,Haruhito Watanabe,Hiroyuki Kusumoto
Keio University

Abstract In this paper,we discribe the scheme to construct virtual broadcast link for uni-directional network to solve some routing problems with unidirectioanl links. We also indicate validity of our scheme with simulation on LAN. When we deploy unidirectional link in the Internet, general assumption of bidirectional link is problem. This disables existing routing protocol work properly. For example, routing protocols don't work in the networks which include any uni-directional links. Our scheme provides reverse connectivity of unidirectional link by IP tunneling. Further, it also achieves same function as general broadcast link, sending datalink packet originated by receiving-only node connected to uni-directional link via the node with sending ability.

1 背景

インターネットで使用されている通信媒体のほとんどは、双方向性を備えている。一方、WWWやFTPによるダウンロードなどのサービスでは、クライアントからの要求に比べてサーバから送信されるデータ量が非常に多い[1]。そのためIPパケットの多くはサーバからクライアントへ流れる傾向にある。従来の回線は双方向の帯域が対称であり、このような傾向にあるトラフィックには適さない。これに適する通信路として、衛星回線やCATV網など片方向の回線(Uni-Directional-Link, 以下UDLと呼ぶ)をインターネットで利用することが考えられている。UDLと既存の双方向回線との併用により、クライアントからの要求に双方向回線、サーバからの応答に片方向回線を利用するなど、現在のトラフィック傾向に適したネットワークを構築できる。

しかし、現在の経路制御技術は通信路の双方向性を前提としている。このため2つの問題が生じる。第一にUDLを含むネットワーク上で動的な経路制御プロトコルが動作しない。経路制御プロトコルはルータ間で経路情報を相互に交換することにより動作している。しかし、UDL上では経路情報が単一方向にしか伝送されない。第二に、UDL上で受信だけを行うノード(以下Receiverと呼ぶ)から同一ネットワーク上の他のノードへ、ネットワーク層での到達性がない。

これらの問題を解決するために、トンネリングを用いてReceiverからUDLに送信する機能を持つノード(以下Feederと呼ぶ)への接続性を仮想的に実現する方法が提案されている[2]。最新の提案では、トンネリングに加え、Receiverからのデータリンクパケットを Feederに送信させ、UDL上に仮想のブロードキャストリンクを構築する。これにより、ネットワーク層のソフトウェアや経路制御プロトコルが、通常のブロードキャストリンクと同じようにUDLを使用できる環境を構築する。しかし、この手法はまだ実装例が少ない。

本論文では、この手法について改善点と新しい提案を示し、実装によりその有効性を評価する。2章では、インターネットでUDLを使用する際の問題について述べる。3章では、UDLの経路制御について、現在考えられているその他の手法を述べる。4章では、仮想ブロードキャストリンクを用いた解決法と、その設計を述べる。5章では、実装について述べる。6章では、LAN上でのシミュレーションを用い、本機構のUDLを含むネットワークでの動作例と実測した性能を述べる。7章ではまとめを行い、今後の課題を示す。

2 インターネットにおける片方向通信路の問題点

片方向の通信媒体の例として、衛星回線やケーブルテレビ網がある。これらには受信と送信の両方を行えるノード (Feeder) と、受信だけを行うノード (Receiver) が存在する。Feeder はその他の Feeder もしくは Receiver への送信とブロードキャストを行う。これに対し、Receiver は UDL を受信専用で利用する。つまり、UDL は片方向のブロードキャストリンクである。Receiver は、外部との通信に UDL を用いられないので、データの送信に双方向回線 (Bi-directional-link, 以下 BDL と呼ぶ) を利用する。従って、UDL に接続する Feeder 及び Receiver のネットワークトポロジは図 1 のようになる。このようなネットワークでは次のような問題が生じる。

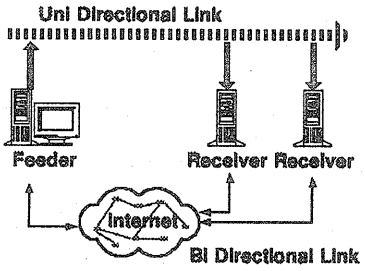

図 1: 片方向通信路を含むネットワークのトポロジ

2.1 動的な経路制御プロトコル

UDL を含むネットワーク上では、動的な経路制御プロトコルが動作しない。ここでは経路制御プロトコルの代表的な例として RIP2 と OSPF を取り上げ、動作しない理由を示す。

RIP2 などの距離ベクトル型経路制御プロトコルは、一定時間ごとに各インタフェースから到達可能情報をブロードキャスト (マルチキャスト) する。RIP2[5] ではこれを response メッセージと呼び、30 秒ごとに送信する。またメッセージは到達可能先とそこまでの距離を含み、メッセージを受け取ったルータはその情報を元に、その先の経路を獲得する。ここで、ルータ A が隣接ルータ B への経路を利用するためには、事前にルータ B から到達可能情報を受け取る必要がある。UDL 上では、データは Feeder から Receiver への片方向にしか流れないため、Feeder は Receiver からの到達可能情報を受け取れず、UDL を用いた経路を使用しない。したがって RIP は正しく動作しない。

OSPF などリンクステート型の経路制御プロトコルでは、全てのルータは自律システムのトポロジが記述された同一のデータベースを保持する。このデータベースを構成する要素は個々のルータの状態であり、使用可能なインタフェース、到達可能な隣接ルータ等を含む。OSPF[6] は隣接ルータとのコネクションの確立、維持に HELLO プロトコルを使用する。特にブロードキャス型ネットワークでは、HELLO プロトコルをマルチキャスト配送することで動的に隣接ルータを発見する。また非ブロードキャスト型ネットワークでは、ユニキャストにより各々の OSPF ルータと指定ルータとの間にコネクションを確立する。HELLO プロトコルは一定間隔ごとにルータの持つ全てのインタフェースから送信され、HELLO メッセージを受け取った先のルータ ID 等を含んでいる。ルータは受け取った HELLO メッセージ内に隣接ルータとして自分のルータ ID を発見すると、双方向の到達性を確認し相手とのコネクションを確立する。しかし UDL では、Receiver から Feeder へ HELLO メッセージを配信できないため、Feeder と Receiver の間にコネクションを確立できない。また同じ理由から、Receiver と指定ルータの間でもコネクションを確立できない。したがって、UDL はトポロジ情報のデータベースに反映されず、OSPF は期待通りには動作しない。

2.2 ネットワーク層における到達不可能性

同一ネットワークのノード間でのデータの送受信は、直接配送で行う。現在の経路制御技術は、直接配送を間接配送よりも優先する [4]。通常、Receiver は外部への送信に BDL を利用するが、UDL 上の終点 IP アドレスに対しては BDL による間接配送よりも UDL による直接配送が優先される。したがって、Receiver は UDL 上の終点 IP アドレスへ UDL 経由で配送しようとし、失敗する。このため、Feeder から Receiver へ、ICMP エコー要求を使用して到達性を確認できないなどの問題が生じる。

2.3 データリンクアドレス解決

マルチアクセスネットワークでは、各インタフェースは一意に識別可能なデータリンクアドレスを持つ。IEEE802LAN 標準化委員会ではデータリンクアドレスとして MAC アドレスを定義しており、現在多くのネットワークインタフェースがこれを使用している。UDL インタフェースもこれに従い MAC アドレスを持つ製品が広まりつつある。このようなネットワークでは、パケットの送信に宛先の MAC アドレスを知る必要がある。しかし、UDL 上では Receiver から Feeder へデータリンクアドレスを伝える手段がないため、データリンク上のアドレス解決プロトコル (ARP) が機能しない。これを解決するため、Feeder が Receiver のデータリンクアドレスを知る機構が必要である。

2.4 通信路の状態検知

UDL では接続するノードが通信路の障害を検知することが難しい。そのため、経路制御プロトコル

をUDL上で動作させた場合、通信路の使用不能を検出するまでに時間がかかる。また、静的に経路制御をした場合、手動で障害を検知し経路を修正する必要がある。この問題を回避するために、UDLの回線状態を検出し、インタフェースを制御する機構が必要である。

3 片方向通信路上における動的経路制御

UDLを含むネットワークで動的な経路制御をするために、現在までに提案されている手法を述べる。

3.1 既存プロトコルの変更

まず、既存の経路制御プロトコルを改変する方法がある。RIPを改変する手法[7]では、ReceiverはFeederへのresponseメッセージをFeederのBDLインタフェース宛に送信する。また、OSPFを改変する手法[8]では、ReceiverはHELLOメッセージをFeederのBDLインタフェース宛に送信する。確立したコネクションは、片方向の接続性としてOSPFのトポロジデータベースに反映される。しかし、これらの手法では自律システム内のルータ全てにおいて変更された経路制御プロトコルが動作しなければならない。したがって改変のためのコストが高く、移行に長い期間が必要である。

3.2 トンネルを用いた解決法

トンネルを用いた方法では、BDLを利用しトンネリングによってReceiverからFeederへパケットを送信する。RIPのresponseメッセージやOSPFのHELLOメッセージ,link-state広告をトンネルを用いて送信することで、UDLを含むネットワーク上で既存の動的な経路制御プロトコルを改変なしに使用できる。この手法は経路制御プロトコルを改変する手法と比較して次の点で優れている。まず、複雑な変更が必要ないので、3.1節で述べた経路制御プロトコルの改変に比べ実装が容易である。また、在来のネットワークに片方向回線を付加する場合、UDLに接続するノードだけに変更を加えればよい。

この手法は現在までにいくつか考えられているが、最新の提案がInternet-Draft[2]として議論されている。本研究ではこれを元に設計と実装を行ない、その改善点と新しい提案を示した。

4 仮想ブロードキャストリンクの設計

4.1 本機構で提供する機能

本機構は、2章で論じた経路制御の問題を解決するため、トンネリングを用いUDL上に仮想ブロードキャストリンクを構築する。Feeder及びReceiverがブロードキャストリンクに接続していた場合、通信路には以下の6つの機能が必要である。

(1) FeederからReceiverへのパケットの送信
(2) ReceiverからFeederへのパケットの送信
(3) FeederからFeederへのパケットの送信
(4) ReceiverからReceiverへのパケットの送信
(5) Feederからのブロードキャスト,マルチキャスト
(6) Receiverからのブロードキャスト,マルチキャスト

Feederからの送信である(1), (3), (5)の機能は既存のUDLで提供されている。したがって、本機構では(2), (4), (6)の機能を提供する。また、MACアドレスを持つUDLインタフェースが広まりつつあることから、UDLインタフェースはデータリンクヘッダによってパケットを識別し、データリンクアドレスにはMACアドレスを使用するものとする。

4.2 トンネル

トンネルにより、BDLを利用してReceiverからFeederへのデータの送信路を提供する。カプセル化方式には、GRE (General Routing Encapsulation)[3]を用いる。既存の提案では、ReceiverはUDLに接続するFeeder全てにトンネルを設定する。しかしこの方法では、Feederの数をNとすると、ReceiverごとにN個のトンネルを設定する必要があり、規模性に欠ける。そこで本設計では、ReceiverはDefault Feederとして選択した1つのFeederへのトンネルだけを使用する。既存の提案では、Default Feederは、Receiverから他のReceiverへ送信する時, Receiverがブロードキャスト, マルチキャスト送信を行う時のトンネル先と考えられている。しかし本設計では、Feederへの送信を含めてReceiverからUDLに送信する全てのデータをDefault Feederに配送する。Default Feederの選出法については後述する。

一般的なトンネリングは、ネットワーク層のパケットを仮想のデータリンクにより送り届ける。しかし本手法では、Receiverはデータリンク層のパケットをトンネルする。Default Feederは、Receiverから受け取ったパケットを宛先IPアドレスにより判別し、UDLに中継する。Default FeederがUDLに送信する際、パケットにはデータリンクヘッダが必要である。データリンクヘッダの送信元アドレスは、送信を行うReceiverが持つUDLインタフェースのMACアドレス、宛先アドレスは、送信先UDLインタフェースのMACアドレスである。このようなデータリンクヘッダは、パケットを中継するDefault Feederに比べ送信元のReceiverのほうがより簡単に作成できる。そこで、Receiverにおいてデータリンクパケットをカプセル化し、中継を行うDefault Feederは脱カプセル化したパケットをそのままUDLに送信する。Receiverでのカプセル化後のパケットを図2に示す。以上は4.1節で示したブロードキャストリンクに必要な機能の(2)にあたる。

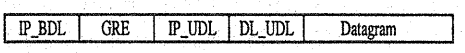

図2: カプセル化されたパケット

4.3 ブロードキャストエミュレーション

Default Feeder はトンネルからパケットを受け取り、脱カプセル化する。次に IP ヘッダの宛先で判別し、次の処理を行う。

(1) 宛先が自分であった場合

UDL インタフェースの input queue にこれを追加する。

(2) 宛先が UDL 上の他のノードであった場合

データリンクパケットをそのまま UDL に送信する。

(3) 宛先がブロードキャストアドレスの場合

データリンクパケットをそのまま UDL に送信すると同時に、このパケットのコピーを UDL インタフェースの input queue に追加する。

(4) 宛先がマルチキャストアドレスの場合

データリンクパケットを UDL に送信し、また自分が宛先マルチキャストアドレスグループに入っていた場合には、UDL インタフェースの input queue にこれを追加する。

以上により、Receiver から Receiver への送信, Receiver からのブロードキャスト, Receiver からのマルチキャストが実現される。これは 4.1 節で示したブロードキャストリンクに必要な機能の (4) と (6) にあたる。

4.4 動的なトンネルの制御

Receiver が Default Feeder にトンネルを設定するために、Default Feeder が持つ BDL インタフェースの IP アドレスを知る必要がある。この設定を手動で行った場合、管理に大きな手間がかかる。そこで、この作業を自動化しトポロジの変化に動的に対応するために、DTCP(Dynamic Tunneling Configuration Protocol)[2] を用いる。また、本設計ではトンネルを Default Feeder の 1 つに設置するため、既存の提案に比べ DTCP をより簡略化して使用する。

DTCP では、Feeder は自分が持つ全ての BDL インタフェースの IP アドレスを UDL に定期的にアナウンスする。これを HELLO メッセージと呼ぶ。図 3 に HELLO メッセージのパケットフォーマットを示す。

Version	Com	Interval	Sequence	
res F	IP Vers	Tunnel Type	Number of FBIP	reserved
Feed UDL IP addr(FUIP)				(32/128 bits)
Feed BDL IP addr(FBIP 1)				(32/128 bits)
. . . . .				
Feed BDL IP addr(FBIP n)				(32/128 bits)

図 3: HELLO メッセージのパケットフォーマット

Com フィールドは JOIN もしくは LEAVE のいずれかの値を含む。UDL に送信可能な Feeder は JOIN メッセージを送信し、Interval フィールドにこの送信間隔を挿入する。Feeder が UDL への送信を止める場合、LEAVE メッセージを用いてリンクのダウンを報告する。JOIN メッセージを受け取った Receiver は、FBIP に含まれるアドレスの中から最適なものを選び FUIP のトンネル先に設定し、タイマを初期化する。トンネルがすでに設定されていた場合、sequence フィールドの値を直前に受け取ったメッセージと比較し、これが同一でないことを確認してタイマを初期化する。LEAVE メッセージを受け取った場合、Receiver は UDL インタフェースをダウンさせ、該当するトンネルを閉じる。

既存の提案では、Receiver は全ての Feeder と Default Feeder へのトンネルを制御する。しかし、本設計では JOIN メッセージを受け取った Feeder の中から 1 つを Default Feeder に選出し、これへのトンネルだけを制御する。Default Feeder との通信が途切れる場合は、新しい Default Feeder を選ぶ。

Default Feeder の選出法はこれまで提案されていない。Default Feeder には Receiver から最も効率的にパケットを送信できるものがよいが、その算出は困難である。したがって、本設計ではこれを最も早く JOIN メッセージを受け取った Feeder とする。

4.5 データリンクアドレス解決

データリンクアドレスの解決は 2 つの方式が考えられる。1 つは送信側が送信時に受信側に対しデータリンクアドレスを要求する手法で、上位層アドレスとデータリンクアドレスを動的に結合する。また、IP アドレス及びデータリンクアドレスの両方を選択する自由がある場合、これらを計算によって対応づける直接マップによる解決法がある [9]。これらのうちどちらを使用すべきかは通信路の性質による。本機構は UDL を含むネットワークにおけるネットワーク層の経路制御問題を解決するものであり、データリンクアドレスの解決は扱う問題の範囲外である。

4.6 その他のブロードキャスト・マルチキャストエミュレーションとの比較

ATM では、LANE(LAN Emulation)[10] と呼ばれる機構がブロードキャストリンクをエミュレーションする。この機構では、LES(LAN Emulation Server) が IP アドレスと ATM アドレスを照会し、MARS (Multicast Address Resolution Server) が IP マルチキャスト、IP ブロードキャストアドレスと ATM アドレスを照会する。この手法を UDL 上で用いた場合、次のような機構が設計できる。Receiver は、LES に問い合わせ、UDL 上のノードへのトンネル先を獲得する。また、MARS に問い合わせ、マルチキャスト, ブロードキャストアドレスと配信すべきノードの BDL インタフェースのリストを照会する。しかし、この手法を用いた場合 Receiver は LES や MARS と頻繁に通信する必要があり、これらが ATM のように互いに高速に通信できる環境にあるとは限らない。また、規模性の問題がある。UDL に多くのノードが

接続した場合、マルチキャストやブロードキャスト配信に際し、Receiver は多数のノードにトンネルを通してパケットを送信する必要がある。ATM ではポイント-マルチポイント接続が提供されているが、トンネルでこの機能を実現することは難しい。したがって、LANE 方式は UDL 上にブロードキャストリンクを構築する手法として適さない。

5 実装

本機構は GRE トンネリング、ブロードキャストエミュレーション、DTCP の 3 つの部分から構成される。実装には FreeBSD2.2.6-RELEASE を用いた。GRE トンネリングとブロードキャストエミュレーションは、カーネル内部に実装した。また、DTCP はユーザ空間で動作するプログラムとして実装した。

5.1 GRE トンネル

GRE トンネリング機構は、Receiver でのデータリンクパケットのカプセル化と、Feeder での脱カプセル化を行う。

まず、Receiver のカプセル化について述べる。Receiver のトンネリング機構は、トンネル先の IP アドレスをカーネル内に保持している。既存の提案では、UDL に送信する IP パケットの宛先 IP アドレスごとにトンネル先を選択する。しかし、本設計では宛先 IP アドレスにかかわらず Default Feeder へのトンネルを使用するため、既存の提案に比べ処理のコストが少なくなっている。

UDL に送信するデータリンクパケットは、トンネリング機構に渡される。トンネリング機構は GRE ヘッダを付加し、それに IP ヘッダを付加する。IP ヘッダの宛先フィールドはトンネル先の IP アドレスであり、プロトコルフィールドは GRE である。カプセル化後の IP パケットは、BDL を使用し Feeder の BDL インタフェースへ送信される。

次に、Feeder の脱カプセル化について述べる。既存の提案では、脱カプセル化とパケットの判別をデータリンク層で行う。しかし、本実装では Feeder はこれらの処理を IP 層で行っている。あるパケットがトンネルを通して送られたことを判別するためには、IP ヘッダのプロトコルフィールドを参照する。この処理はデータリンク層に比べ IP 層で行うほうがより合理的であり、オペレーティングシステムへの変更が少なくてすむ。

IP ヘッダのプロトコルフィールドが GRE の場合、パケットはトンネリング機構に渡される。トンネリング機構は IP ヘッダと GRE ヘッダを取り除き、データリンクパケットをブロードキャストエミュレーションの判別機構に送る。Feeder 及び Receiver のトンネリング機構でパケットがどのように扱われるかを図 4 に示す。

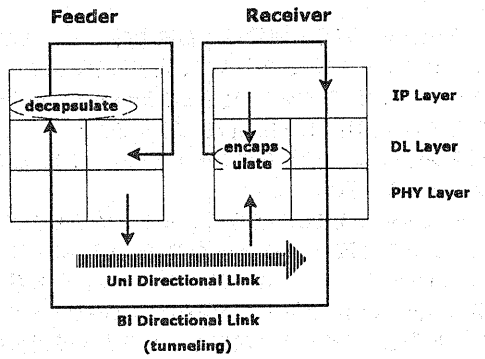

図 4: カプセル化と脱カプセル化

5.2 ブロードキャストエミュレーション

Feeder は、ブロードキャストエミュレーションのために、カーネル内に以下のパラメータを保持する。

・UDL に接続するインタフェースの IP アドレス
・UDL ネットワークのネットマスク
・UDL ネットワークのブロードキャストアドレス
・UDL のインタフェース構造体へのポインタ

トンネリング機構が脱カプセル化したデータリンクパケットを、内部の IP ヘッダにより判別する。宛先アドレスが Feeder の IP アドレスに一致する場合、通常の受信処理を行う。宛先アドレスが UDL ネットワークのブロードキャストアドレスに一致する場合、データリンクパケットを UDL のインタフェース構造体内部のポインタが指す送信ルーチンに渡し、そのまま UDL に送信する。送信ルーチンはパケットをコピーし、loopback インタフェースに送る。宛先がマルチキャストアドレスの場合、データリンクパケットをそのまま UDL に送信し、自分がそのマルチキャストグループに入っている場合には、パケットのコピーを loopback インタフェースに送る。上述のどれにもあてはまらなかったパケットは、UDL に接続するその他のノード宛とみなし、そのまま UDL に送信する。データリンクパケットを UDL インタフェースからそのまま送信するために、UDL への送信ルーチンは、転送するデータリンクパケットにヘッダを付加しないように変更を加えた。この判別のために、UDL にそのまま送信するパケットにはバッファごとにフラグを ON にする。トンネルから受け取ったパケット全てにこのフラグを立てると処理が簡略化される。しかし、ICMP エコー応答では受け取ったエコー要求と同じバッファを使用するので、このようなパケットのバッファのフラグを ON にすると、データリンクヘッダが付加されず正常に送信されない問題が発生する。

5.3 DTCP

Feeder の DTCP 機構は、初期設定として UDL インタフェースの IP アドレス, 全ての BDL インタフェースの IP アドレス, HELLO Interval の値を保持している。HELLO Interval ごとに JOIN メッセージを送信し、ユーザからのシグナルによって LEAVE メッセージを送信する。

Receiver の DTCP 機構は、はじめに JOIN メッセージを受け取った Feeder を Default Feeder に選ぶ。JOIN メッセージの FBIP フィールドから最も早く検出した IP アドレスを Default Feeder へのトンネル先 IP アドレスに設定し、リンクを使用可能にする。Default Feeder から LEAVE メッセージを受け取った場合はリンクを使用不可にする。

6 評価

6.1 実験環境

本機構を評価するため、図5に示すネットワークを構築した。このネットワークは、片方向通信路を含む

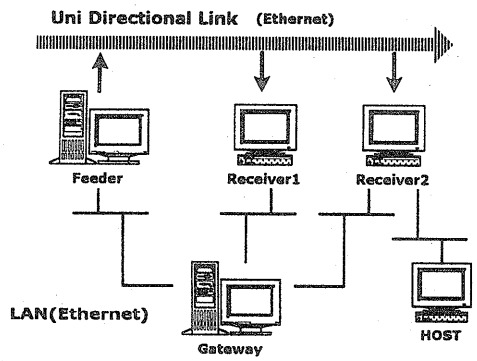

図 5: 評価を行ったトポロジ

ネットワークをエミュレートする Ethernet の LAN である。各ノードの OS には FreeBSD2.2.6-RELEASE を用いた。経路制御プロトコルには、RIP2 と OSPFv2 を用い gated Release3.5 で動作させた。評価項目は、次の 2 つである。

・経路制御プロトコルにより、UDL を使用した経路を含む経路表が作成される

・Receiver から他の UDL 上の終点 IP アドレスへ到達性がある

6.2 実験結果

RIP2 を動作させたところ、UDL を使用した経路を含む経路表が Feeder, Receiver において作成された。OSPFv2 の場合も同様に、UDL を使用した経路を含む経路表が作成された。Feeder から HOST へ IP パケットを送信した場合、UDL を経由し Receiver2 によってフォワーディングされ、HOST に到達することを確認した。

また ping プログラムを用いて、Feeder から Receiver, Receiver から Receiver, Feeder からのブロードキャスト, Receiver からのブロードキャストの到達性を確認した。

7 むすび

本論文では、片方向通信路をインターネットで使用する際に生じる問題点を述べ、既存の経路制御プロトコルが片方向通信路を含むネットワークでは正しく動作しないことを示した。これを解決するため、トンネルを用いて片方向通信路の復路となる接続性を提供する手法を示し、既存の設計に新しい提案を行った。また、これを実装し、Ethernet でのシミュレーションを用いて、本機構を使用した場合に動的な経路制御プロトコルが正常に動作することを確認した。

しかし、本機構を用いた場合、UDL が通常の BDL と同じ様に扱われるため、利用や運用の仕方によってトンネルに過剰なパケットが流れる恐れがある。本機構の効率的な利用法が、今後の課題である。

謝辞 本研究に対して貴重な助言を頂いた (株)NTT サテライトコミュニケーションズの泉山英孝氏、竹井淳氏、(株) ソニーの藤井昇氏をはじめとする WIDE プロジェクトの方々に感謝致します。また、本研究を進める上で終始惜しみない御助力をいただきました慶應義塾大学政策・メディア研究科の西田視磨氏に深く感謝致します。

参考文献

[1] 前田 亮, 植村 俊亮, WWW におけるマルチメディアファイルの容量分析, 第 9 回データ工学ワークショップ論文集, March 1998

[2] E.Duros, W.Dabbous, H.Izumiyama, N.Fujii, and Y.Zhang, A Link Layer Tunneling Mechanism for Unidirectional Links, Internet-Draft,March 1998

[3] S.Hanks, T.Li, D.Farinacci, P.Traina, Generic Routing En apsulation, RFC1701,October 1994

[4] R.Braden, Requirements for Internet Hosts – Communication Layers, RFC1122,October 1989

[5] G.S.Malkin, RIP Version 2 Carrying Additional, Information,RFC1388,January 1993

[6] J.Moy, The OSPF Version 2, RFC1583,March 1994

[7] E.Duros, C.Huitema, Handling of unidirectional links with RIP, Internet-Draft,Novemver 1997

[8] E.Duros, Handling of unidirectional links with OSPF, Internet-Draft,March 1996

[9] David C. Plummer, An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, RFC826,November 1982

[10] T.Smith, G.Armitage, IP Broadcast over ATM Networks, RFC2226,October 1997